AMENDMENTS TO THE CLAIMS

1. (Canceled)

2. (Canceled)

3. (Canceled)

4. (Currently amended) In a data processing system having a least one processor for running tasks and resources available to tasks, a method comprising the steps of: ~~The method recited in Claim 1 wherein the step of logically partitioning tasks into groups of tasks that utilize the same resources further comprises the steps of~~:

logically partitioning tasks into groups of tasks that utilize the same resources, including initially placing each task in its own group; ~~and~~

subsequently combining groups of tasks into a merged group wherein each task in the merged group allocates a common resource when run[.] ;

preemptively scheduling the groups of tasks to be run such that each group of tasks is given a time slot in a cycle in which its tasks may run on the processor; and

for each group, non-preemptively scheduling tasks to be run within each group during the time slot allocated to the group.

5. (Canceled)

6. (Canceled)

7. (Canceled)

8. (Original) In a data processing system having at least one storage device for storing modules of code and at least one processor for running tasks, and resources that may be allocated by tasks, wherein running each task involves allocating at least one resource, a method comprising the steps of:

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

providing a module and resource dependency list for each associated module of code, wherein each module and resource dependency list lists interdependent modules of code of the associated module of code and resources allocated by the associated module of code;

generating a task dependency list for each task by taking a logical union of modules and resources listed in the module and resource dependency lists of modules and resources that are candidates to be called when the task is run on the processor;

examining the task dependency lists to logically partition the tasks into groups of interdependent tasks; and

preemptively scheduling the groups of tasks to be run such that each group of tasks is given a time slot in a cycle in which its tasks may run on the processor; and

for each group of tasks, non-preemptively scheduling tasks to be run within each group during the time slot allocated to the group.

9. (Original) The method recited in Claim 8, further comprising the step of storing a group list for each group that holds identifying information for tasks included in the group.

10. (Original) The method recited in Claim 8, further comprising the step of storing status information for each group indicating the group has a task that is running and holding identifying information about any task that is running.

11. (Canceled)

12. (Canceled)

13. (Currently amended) In a data processing system having at least one processor for running tasks and resources that may be allocated by said tasks, wherein said processor runs an operating system, a computer-readable storage medium holding the operating system, said operating system performing the steps of ~~The computer-readable storage medium of Claim 12 wherein the operating system further performs the steps of~~:

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

logically partitioning tasks into groups of interdependent tasks, including initially placing each task in its own group[;] ,wherein said tasks are to be run non-preemptively, said tasks allocating the same resources, and said tasks that do not allocate the same resources being placed in separate groups; ~~and~~

subsequently combining groups of tasks into a merged group wherein each task in the merged group allocates a common resource[.] ;

preemptively scheduling the groups of tasks to be run such that each group of tasks is given a time slot in a cycle in which its tasks may run on the processor; and

for each group, non-preemptively scheduling tasks to be run within each group during the time slot allocated to the group.

14. (Canceled)

15. (Original) In a data processing system having at least one storage device for storing modules of code, system resources that may be allocated by tasks, and at least one processor for running tasks wherein running each task involves allocating at least one resource, a computer-readable storage medium holding an operating system for performing the steps of:

providing a module and resource dependency list for each associated module of code, wherein each module and resource dependency list lists interdependent resources of the associated module of code;

generating a task dependency list for each task by taking a logical union of resources listed in the module and resource dependency lists of resources that are allocated when the task is run on the processor;

examining the task dependency lists to logically partition the tasks into groups of interdependent tasks;

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

preemptively scheduling the groups of tasks to be run such that each group of tasks is given a time slot in a cycle in which its tasks may run on the processor; and

for each group of tasks, non-preemptively scheduling tasks to be run within each group during the time slot allocated to the group.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100